UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/916,900	07/26/2001	Gowri Rajaram	UTL 00073	7631

7590 04/22/2004

Kyocera Wireless Corp.
Attn: Patent Department
10300 Campus Point Drive
San Diego, CA 92121

EXAMINER	
NGUYEN, JOSEPH D	
ART UNIT	**PAPER NUMBER**
2683	6

DATE MAILED: 04/22/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No. 09/916,900	Applicant(s) RAJARAM ET AL.	
	Examiner Joseph D Nguyen	Art Unit 2683	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *26 July 2001*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-57* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-57* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *26 July 2001* is/are: a)☐ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date *5/1/26/04*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

Claim Rejections - 35 USC § 102

1. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

2. Claims 1-3, and 31 are rejected under 35 U.S.C. 102(b) as being anticipated by Kay (5,930,704).

Regarding claim 1, Kay discloses in a wireless communications device, a method for updating system software stored in memory (abstract, fig. 1), the method comprising:

a) storing system software for the wireless device in a plurality of current code sections (fig. 17-20, col. 2 lines 5-53);

b) receiving a new code section (abstract, fig. 10, 24, col. 2 lines 5-53);

c) arranging the new code section with current code sections to form updated system software for the wireless device (fig. 17-20, col. 2 lines 5-53); and,

d) executing the updated system software (fig. 21-22, col. 2 lines 5-53, and col. 15 lines 15-17).

Regarding claim 2, Kay further discloses the method of claim 1 further comprising:

a) identifying a first code section for updating (fig. 28-29, 33-34, col. 2 lines 5-53); and,

b) wherein arranging the new code section with current code sections to form updated system software includes replacing the first code section with the new code section (fig. 17-33, col. 2 lines 5-53, col. 16 lines 30-59, and col. 22 line 58 thru col. 23 line 34).

Regarding claim 3, Kay further discloses the method of claim 2 wherein executing the updated system software includes using the new code section (new version number) in executing the updated system software (abstract, fig. 21-22, col. 2 lines 5-53, and col. 15 lines 15-17).

Regarding claim 31, Kay discloses in a wireless communications device, a method for updating system software stored in memory (abstract, fig. 1), the method comprising:

a) storing system software for the wireless device in a plurality of current code sections (fig. 17-20, col. 2 lines 1-53, and col. 14 lines 1-35);

b) receiving new code sections via a wireless communications device air interface (fig. 27-29, col. 2 lines 1-54, and col. 21 line 33 thru col. 24 line 8);

c) storing the new code sections in a memory file system section (fig. 27-29, col. 2 lines 1-54, and col. 21 line 33 thru col. 24 line 8);

d) identifying current code sections for updating (col. 2 lines 5-27);

e) replacing current code sections with new code sections to form updated system software for the wireless device; and, executing the updated system software (col. 14 line 1 thru col. 18 line 13).

Claim Rejections - 35 USC § 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. Claims 4-30, and 33-57 are rejected under 35 U.S.C. 103(a) as being unpatentable over Kay (5,930,704) in view of Lillich (5,790,856).

Regarding claim 4, Kay further discloses the method of claim 1 with compiling the system software into a library and creating the overwrites the exit module in a copy of the library (col. 17 line 58 thru col. 18 line 13). However Kay does not specifically discloses forming the system software into a first plurality of symbol libraries, each symbol library comprising at least one symbol; and, arranging the first plurality of symbol libraries into a second plurality of code sections.

Lillich teaches forming the system software into a first plurality of symbol libraries, each symbol library comprising at least one symbol (col. 5 line 57 thru col. 7 line 32); and, arranging the first plurality of symbol libraries into a second plurality of code sections (fig. 9, col. 5 line 57 thru col. 7 line 32, col. 12 lines 11-35). Therefore, it would have been obvious to one skilled in the art at the time the invention was made to modify the Kay system with the teaching of Lillich of forming system software into a first plurality of symbol libraries, each symbol library comprising at least one symbol and arranging the symbol libraries into a second code sections (code fragment) in order to identified each fragment by its name, its usage and its location when upgrading the software.

Regarding claim 5, Lillich further discloses the method of claim 4 wherein forming system software into a first plurality of symbol libraries includes each symbol library comprising symbols having related functionality (col. 5 line 57 thru col. 7 line 32).

Regarding claim 6, Kay further discloses the method of claim 5 wherein receiving a new code section includes receiving the new code section via a wireless communications device air interface (abstract, col. 1 lines 51-67, and col. 15 lines 15-17).

Regarding claim 7, Kay further discloses the method of claim 6 wherein forming system software into a symbol library includes forming read-write data for a symbol library (col. 14 line 1 thru col. 15 line 67); and, wherein arranging code sections includes arranging the read-write data in a shared read-write code section (fig. 17-28, col. 2 lines

5-26, and col. 14 line 1 thru col. 15 line 67). However, Kay does not specifically disclose forming system software into a first plurality of symbol libraries includes forming data for a plurality of symbol libraries; and, wherein arranging the first plurality of symbol libraries into a second plurality of code sections includes arranging the data in a shared code section.

Lillich teaches forming system software into a first plurality of symbol libraries includes forming read-write data for a plurality of symbol libraries (fig. 9, col. 5 line 57 thru col. 6 line 67); and, wherein arranging the first plurality of symbol libraries into a second plurality of code sections includes arranging the read-write data in a shared read-write code section (fig. 7-9, col. 10 line 66 thru col. 11 line 34). Therefore, it would have been obvious to one skilled in the art at the time the invention was made to modify the Kay system with the teaching of Lillich of symbol libraries in order to determine which patch library code fragment are linked to the retrieved import library code fragments when updating the system software.

Regarding claim 8, Kay further discloses the method of claim 7 wherein receiving a new code section includes receiving an updated read-write code section (fig. 17, col. 2 lines 5-26, col. 14 line 1 thru col. 15 line 67).

Regarding claim 9, Kay further discloses the method of claim 8 wherein identifying a first code section for updating includes identifying the read-write code section (fig. 17-28, col. 2 lines 5-26, and col. 14 line 1 thru col. 15 line 67); wherein arranging the new code section with current code sections to form updated system

software includes replacing the read-write code section with the updated read-write code section (fig. 17-27, col. 2 lines 5-26, and col. 14 line 1 thru col. 16 line 9); and wherein executing the updated system software includes using the updated read-write code section in executing of the updated system software (fig. 21-28, col. 2 lines 5-26, and col. 14 line 1 thru col. 16 line 9).

Regarding claim 10, Kay further discloses the method of claim 9 wherein arranging the symbol libraries into a code sections includes starting at the start of code sections (fig. 25-34, col. 2 lines 5-26, col. 7 lines 41-52, and col. 17 line 54 thru col. 18 line 13); the method further comprising:

a) storing the start of code sections at corresponding start addresses (fig. 16, col. 15 line15 thru col. 17 line 24), and

b) maintaining a code section address table cross-referencing code section identifiers with corresponding start addresses (fig. 20, 33, col. 16 line 46 thru col. 17 line 24). However, Kay does not specifically disclose wherein arranging the first plurality of symbol libraries into a second plurality of code sections includes starting symbol libraries at the start of code sections.

Lillich teaches wherein arranging the first plurality of symbol libraries into a second plurality of code sections includes starting symbol libraries at the start of code sections (fig. 7-11, col. 5 line 57 thru col. 6 line 67). Therefore, it would have been obvious to one skilled in the art at the time the invention was made to modify the Kay system with the teaching of Lillich of symbol libraries in order to determine which patch

library code fragment are linked to the retrieved import library code fragments when updating the system software.

Regarding claim 11, Lillich further discloses the method of claim 10 wherein arranging the first plurality of symbol libraries into a second plurality of code sections includes arranging each symbol to be offset from its respective code section start address (fig. 9, col. 12 lines 24-35, col. 12 lines 3-13, and col. 16 lines 30-40); and the method further comprising: maintaining a symbol offset address table cross-referencing symbol identifiers with corresponding offset addresses, and corresponding code section identifiers (fig. 9, 11). Therefore, it would have been obvious to one skilled in the art at the time the invention was made to modify the Kay system with the teaching of Lillich of symbol to be offset and offset address table cross-referencing symbol identifiers in order to determine which patch library code fragment are linked to the retrieved import library code fragments when updating the system software.

Regarding claim 12, Lillich further discloses the method of claim 11 wherein arranging the first plurality of symbol libraries into a second plurality of code sections includes:

a) arranging the code section address table in a first table code section (fig. 4, 7-9, col. 9 lines 19-30, col. 13 line 3 thru col. 14 line 10).

b) arranging the symbol offset address table in a second table code section (fig. 4, 7-9, col. 9 lines 19-30, col. 13 line 3 thru col. 14 line 10). Therefore, it would have been obvious to one skilled in the art at the time the invention was made to modify the

Kay system with the teaching of Lillich of symbol offset address table in order to determine which patch library code fragment are linked to the retrieved import library code fragments when updating the system software.

Regarding claim 13, Kay further discloses the method of claim 12:

a) wherein receiving an updated code section includes receiving an updated first table code section and an updated second table code section (fig. 20-34, col. 2 lines 5-19, and col. 16 line 30 thru col. 17 line 57);

b) wherein arranging the new code section with current code sections to form updated system software includes replacing the first table code section with the updated first table code section (fig. 20-34, col. 2 lines 5-19, and col. 16 line 30 thru col. 17 line 57), and the second table code section with the updated second table code section (fig. 20-34, col. 2 lines 5-19, and col. 16 line 30 thru col. 17 line 57); and,

c) wherein executing the updated system software includes using the updated first table code section and updated second table code section in executing the updated system software (fig. 20-34, col. 2 lines 5-19, and col. 16 line 30 thru col. 17 line 57).

Regarding claim 14, Kay further discloses the method of claim 13 wherein forming system software and arranging code section, and store the new code section (col. 2 lines 1-40). However, Kay does not specifically disclose wherein forming into a first plurality of symbol libraries includes forming a patch library; wherein arranging the first plurality of symbol libraries into a second plurality of code sections includes arranging a patch library into a patch manager code section; wherein arranging the new

code section with current code sections to form updated system software for the wireless device includes: accessing the patch manager code section; and, invoking the patch library to store the new code section.

Lillich teaches forming into a first plurality of symbol libraries includes forming a patch library (abstract, fig. 8-9); wherein arranging the first plurality of symbol libraries into a second plurality of code sections includes arranging a patch library into a patch manager code section (fig. 6-9, col. 6 line 49 thru col. 7 line 15); wherein arranging the new code section with current code sections to form updated system software for the wireless device includes: accessing the patch manager code section (fig. 6, col. 10 lines 8-16); and, invoking the patch library to store the new code section (import in patch library) (fig. 9, 11, col. 10 lines 40-65). Therefore, it would have been obvious to one skilled in the art at the time the invention was made to modify the Kay system with the teaching of Lillich in order to improve the method of upgrading software.

Regarding claim 15, Lillich further discloses the method of claim 14 wherein invoking the patch library to store the new code section includes invoking the patch library to over-write the first code section with the new code section (abstract, fig. 7, col. 5 line 57 thru col. 6 line 48).

Regarding claim 16, Lillich further discloses the method of claim 15 further comprising: after receiving the new code section, storing the new code section in a memory file system section; wherein arranging the new code section with current code sections to form updated system software includes invoking the patch library to over-

write the first code section with the new code section stored in the memory file system section (fig. 7-9, col. 10 line 66 thru col. 12 line 35).

Regarding claim 17, Lillich further discloses the method of claim 16 wherein receiving a new code section includes receiving an updated patch manager code section (fig. 12, col. 15 lines 9-55); wherein arranging the new code section with current code sections to form updated system software includes replacing the patch manager code section with the updated patch manager code section (fig. 5-12, col. 9 line 31 thru col. 11 line 60); and, wherein executing the updated system software includes using the updated patch manager code section in executing the updated system software (fig. 6-14, col. 6 line 49 thru col. 7 line 47).

Regarding claim 18, Lillich further discloses the method of claim 17 wherein arranging the first plurality of symbol libraries into a second plurality of code sections includes arranging read-write data, the code section address table, and the symbol offset address table in the patch manager code section (fig. 6-14, col. 10 line 8 thru col. 13 line 13); and, wherein receiving an updated patch manager code section includes receiving an updated symbol offset address table, an updated code section address table, and updated read-write data (fig. 6-14, col. 10 line 8 thru col. 13 line 13).

Regarding claim 19, Lillich further teaches the method of claim 17 wherein forming system software into a first plurality of symbol libraries includes forming a symbol accessor code (fig. 6, col. 10 lines 8-65); wherein arranging the first plurality of symbol libraries into a second plurality of code sections includes arranging the symbol

accessor code in the patch manager code section (fig. 6-14); and, the method further comprising: storing the symbol accessor code address at a first location in memory (fig. 7-14); wherein executing the updated system software includes: in response to referencing the first location in memory, accessing the symbol accessor code (fig. 2-9, col. 5 line 57 thru col. 7 line 47); and, invoking the symbol accessor code to calculate the address of a sought symbol using a corresponding symbol identifier, and a corresponding code section identifier (fig. 9-14, col. 12 line 11 thru col. 14 line 10).

Regarding claim 20, Lillich further discloses the method of claim 19 wherein invoking the symbol accessor code to calculate the address of the sought symbol includes accessing the code section address table, and the symbol offset address table to calculate the address of the sought symbol (fig. 4-14, col. 12 line 36 thru col. 16 line 67).

Regarding claim 21, Lillich further discloses the method of claim 20 wherein receiving an updated patch manager code section includes receiving an updated symbol accessor code (fig. 6-14); wherein replacing the patch manager code section with updated patch manager code section includes replacing the symbol accessor code with updated symbol accessor code (fig. 6-14); and, wherein executing the updated system software includes using the updated symbol accessor code in executing the updated system software (fig. 6-14).

Regarding claim 22, Lillich further discloses the method of claim 21 further comprising: receiving an updated symbol accessor code address (fig. 6-14, col. 10 line

8 thru col. 16 line 67); storing the updated symbol accessor code address in the file system section (fig. 9-14, col. 12 lines 11-63); replacing the symbol accessor code address in the first location in memory with updated symbol accessor code address from the file system section (fig. 6-9, col. 10 line 8 thru col. 11 line 60); and, wherein executing the updated system software includes using the updated symbol accessor code address in executing the updated system software (fig. 6-9, col. 10 line 8 thru col. 12 line 63).

Regarding claim 23, Lillich further discloses the method of claim 22 wherein arranging the first plurality of symbol libraries into a second plurality of code sections includes arranging the symbol accessor code address in the patch manager code section (fig. 4-9, col. 9 line 19 thru col. 12 line 63); and, wherein replacing the symbol accessor code address in the first location in memory with the updated symbol accessor code address from the file system section includes replacing the symbol accessor code address in the patch manager code section with the updated symbol accessor code address in an updated patch manager code section (fig. 6-9, col. 10 line 8 thru col. 12 line 63).

Regarding claim 24, Lillich further discloses the method of claim 23 wherein executing the system software includes: loading the read-write data, the code section address table, the symbol offset address table, the patch library, symbol accessor code, and a symbol accessor code address from the patch manager code section into read-write volatile memory; and, accessing the read-write data, the code section address

table, the symbol offset address table, patch library, the symbol accessor code, and the symbol accessor code address from read-write volatile memory (fig. 6-14).

Regarding claim 25, Lillich further discloses the method of claim 24 wherein storing the start of code sections at corresponding start addresses includes: creating a second plurality of contiguously addressed memory blocks (fig. 6-9, col. 9 line 19 thru col. 12 line 63); identifying each memory block with a corresponding code section (fig. 6-9, col. 9 line 19 thru col. 12 line 63); and, storing code sections in the identified memory blocks (fig. 4-9, col. 9 line 19 thru col. 12 line 63).

Regarding claim 26, Lillich further discloses the method of claim 25 wherein arranging the first plurality of symbol libraries into a second plurality of code sections includes arranging a third plurality of symbol libraries in a first code section (fig. 1-14, col. 9 line 19 thru col. 12 line 63); wherein identifying each memory block with a corresponding code section includes identifying a first memory block with the first code section (fig. 7, col. 10 line 66 thru col. 11 line 38); wherein storing code sections in the identified memory blocks includes storing the third plurality of symbol libraries in the first memory block (fig. 1-9, col. 12 line 11 thru col. 13 line 14); wherein receiving a new code section includes receiving an updated first code section with third plurality of symbol libraries arranged within (fig. 9-14, col. 12 line 11 thru col. 16 line 67); and, wherein arranging the new code section with current code sections to form updated system software for the wireless device includes overwriting the first code section in the first memory block with an updated first code section (fig. 8-9, col. 11 line 38 thru col. 12 line 35).

Regarding claim 27, Lillich further discloses the method of claim 25 wherein arranging the first plurality of symbol libraries into a second plurality of code sections includes arranging a first symbol library in a first code section (col. 5 line 57 thru col. 7 line 32); wherein identifying each memory block with a corresponding code section includes identifying a first memory block with the first code section (fig. 6-9, col. 10 line 66 thru col. 11 line 38); wherein storing code sections in the identified memory blocks includes storing the first symbol library in the first memory block (fig. 6-9, col. 10 line 66 thru col. 11 line 38); wherein receiving a new code section includes receiving an updated first code section with first symbol library arranged within (col. 5 line 57 thru col. 7 line 32); and, wherein arranging the new code section with current code sections to form updated system software for the wireless device includes overwriting the first code section in the first memory block with an updated first code section (fig. 1-14, col. 10 line 8 thru col. 13 line 13).

Regarding claim 28, Lillich further discloses the method of claim 25 wherein arranging the first plurality of symbol libraries into a second plurality of code sections includes sizing the code sections to accommodate arranged symbol libraries (fig. 7-10, col. 10 col. 66 thru col. 14 line 17); and, wherein creating a second plurality of contiguously addressed memory blocks includes sizing memory blocks to accommodate corresponding code sections (fig. 6-9, col. 10 line 8 thru col. 12 line 35).

Regarding claim 29, Lillich further discloses the method of claim 28 wherein arranging the first plurality of symbol libraries into a second plurality of code sections

includes sizing the code sections to accommodate sizes larger than the arranged symbol libraries (fig. 6-10, col. 10 line 8 thru col. 14 line 17).

Regarding claim 30, Kay further discloses the method of claim 4 wherein storing system software for the wireless device in a plurality of current code sections includes storing system software in a second plurality of code sections (col. 14 line 1 thru col. 16 line 60); wherein receiving a new code section includes receiving a second plurality of updated code sections (col. 14 line 1 thru col. 18 line 13); wherein arranging the new code section with the current code section includes replacing the second plurality of current code sections with the second plurality of updated code sections (col. 14 line 1 thru col. 18 line 13); and, wherein executing the updated system software includes using the second plurality of updated code sections in executing the updated system software (fig. 15-34, col. 14 line 1 thru col. 24 line 8).

Regarding claim 33, Kay discloses in a wireless communications device, a software updating system (abstract, fig. 15), the system comprising: a code storage section memory including executable wireless device system software differentiated into a plurality of current code sections (fig. 7-22, col. 14 lines 1-36); a file system section memory for receiving new code sections (col. 2 lines 1-40); arranging new code sections in the code storage section with the current code sections (col.14 line 1 thru col. 18 line 13); and, wherein the arrangement of new code sections with current code sections in the code storage section forms updated executable system software (fig. 22). However, Kay does not specifically disclose a software updating system comprising a patch library.

Lillich teaches a software updating system comprising a patch library (fig. 6-10, col. 10 line 8 thru col. 13 line 13). Therefore, it would have been obvious to one skilled in the art at the time the invention was make to modify the Kay system with the teaching of Lillich of a patch library in order to determine which patch library code to be processed.

Regarding claim 34, Kay further discloses the system of claim 33 further comprising: wherein the file system section receives a first patch manager instruction set (PMRTI) including instruction for arranging the new code section with the current code sections; and wherein the patch library replaces the first code section in the code storage section with the new code section in response to the first PMRTI (col. 15 line 18 thru col. 16 line 9).

Regarding claim 35, this claim is rejected for the same reason as set forth in claim 4.

Regarding claim 36, this claim is rejected for the same reason as set forth in claim 14.

Regarding claim 37, this claim is rejected for the same reason as set forth in claim 5.

Regarding claim 38, this claim is rejected for the same reason as set forth in claim 6.

Regarding claim 39, this claim is rejected for the same reason as set forth in claim 7.

Regarding claim 40, this claim is rejected for the same reason as set forth in claim 13.

Regarding claim 41, this claim is rejected for the same reason as set forth in claim 10.

Regarding claim 42, this claim is rejected for the same reason as set forth in claim 11.

Regarding claim 43, this claim is rejected for the same reason as set forth in claim 13.

Regarding claim 44. Lillich further discloses the system of claim 43 wherein the patch library is a symbol library arranged in a patch manager code section of the code storage section (fig. 8-9)

Regarding claim 45, this claim is rejected for the same reason as set forth in claim 15.

Regarding claim 46, this claim is rejected for the same reason as set forth in claim 17.

Regarding claim 47, this claim is rejected for the same reason as set forth in claim 18.

Regarding claim 48, this claim is rejected for the same reason as set forth in claim 19.

Regarding claim 49, this claim is rejected for the same reason as set forth in claim 20.

Regarding claim 50, this claim is rejected for the same reason as set forth in claim 21.

Regarding claim 51, this claim is rejected for the same reason as set forth in claim 23.

Regarding claim 52, Lillich further discloses the system of claim 51 wherein the patch manager code section includes the first location in memory (fig. 3-5, col.6 lines 49-67).

Regarding claim 53, Kay further discloses the system of claim 52 further comprising: a read-write volatile memory (fig. 15, and 17) to accept the read-write data, the code section address table, the symbol offset address table, the patch library, symbol accessor code, and the symbol accessor code address from the patch manager code section for access during the execution of the system software.

Regarding claim 54, this claim is rejected for the same reason as set forth in claim 25.

Regarding claim 55, this claim is rejected for the same reason as set forth in claim 28.

Regarding claim 56, this claim is rejected for the same reason as set forth in claim 29.

Regarding claim 57, Kay discloses in a wireless communications device, a software updating system (abstract, fig. 3 and 9), the system comprising: a code storage section memory including executable wireless device system software differentiated into a first plurality of symbol libraries arranged in a second plurality of code sections (abstract, fig. 15-22, col. 2 lines 1-53, and col. 14 line 1 thru col. 18 line 37); an airlink interface to receive new code sections, including an updated patch manager code section (abstract, fig. 15-22, col. 14 line 1 thru col. 18 line 37); a file system section memory for storing new code sections received via the airlink interface (fig. 20); wherein the code storage section includes a patch manager code section with a patch library to replace code sections in the code storage section with updated code sections, the patch manager code section further including a code section address table, a symbol offset address table, a symbol accessor code, read-write data, and a symbol accessor code address (abstract, fig. 15-22, col. 14 line 1 thru col. 18 line 37); and, wherein the arrangement of the new code sections, including the updated patch manager code section, with current code sections in the code storage section forms updated executable system software (abstract, fig. 15-22, col. 14 line 1 thru col. 18 line 37). However, Kay does not specifically disclose the update software system includes a patch library, a patch manager.

Lillich teaches the update software system includes a patch library, and a patch manager (abstract, col. 6 lines 7-67). Therefore, it would have been obvious to one

skilled in the art at the time the invention was made to modify Kay system with the teaching of Lillich of a patch manager, and a patch library in order to update the require software in the mobile device to provide the user with new version.

5. Claim 32 is rejected under 35 U.S.C. 103(a) as being unpatentable over Kay (5,930,704) in view of Beasley et al. (5,699,275).

Regarding claim 32, Kay further discloses the method of claim 31 further comprising: forming the system software into a first plurality of symbol libraries including a code section address table (fig. 20-22, col. 16 line 30 thru col. 18 line 12), a symbol offset address table (fig. 33), a symbol accessor code (fig. 18-21, col. 14 line 1 thru col. 17 line 53), and read-write data for a plurality of symbol libraries (col. 7 line 36-67,col. 15 lines 40-67, and col. 17 line 31 thru col. 18 line 13); arranging the code section address table (col. 15 line 40 thru col. 17 line 14), the symbol offset address table, the symbol accessor code (fig. 33), the read-write data (col. 15 line 40 thru col. 1 line 10). However, Kay does not specifically discloses a patch library, and the symbol accessor code address into a patch manager code section; wherein receiving new code sections includes receiving a new patch manager code section; and, wherein replacing current code sections with new code sections to form updated system software for the wireless device includes replacing a current patch manger code section with the new patch manager code section.

Beasley et al. teaches in a wireless communications device, a method for updating system software stored in memory comprises a patch library (patch bank) (abstract, col. 6 line 57 thru col. 7 line 16), and the symbol accessor code address into a patch manager code section (abstract, col. 3 line 10-56); wherein receiving new code sections includes receiving a new patch manager code section (abstract, col. 3 lines 40-56); and, wherein replacing current code sections with new code sections to form updated system software for the wireless device includes replacing a current patch manger code section with the new patch manager code section (col. 3 lines 29-56). Therefore, it would have been obvious to one skilled in the art at the time the invention was made to modify Kay system with the teaching of Beasley of a patch library, patch manager code section in order to update the new software version for mobile device.

6. Any response to this action should be mailed to:

Commissioner of Patents and Trademarks

Washington, D.C. 20231

Or faxed to:

703 308-9051, (for formal communication intended for entry)

Or:

(703) 305-9509 (for informal or draft communications, please label "PROPOSED" OR "DRAFT")

Hand-delivered responses should be brought to Crystal Park II, 2121 Crystal Drive, Arlington. VA. Sixth floor (Receptionist).

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Joseph D Nguyen whose telephone number is (703) 605-1301. The examiner can normally be reached on 7:00 AM to 4:30 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, William Trost can be reached on (703) 308-5318. The fax phone numbers for the organization where this application or proceeding is assigned are (703) 872-9314 for regular communications and (703) 872-9314 for After Final communications.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 306-0377.

Joseph Nguyen

Apr. 15, 2003

WILLIAM TROST
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2600